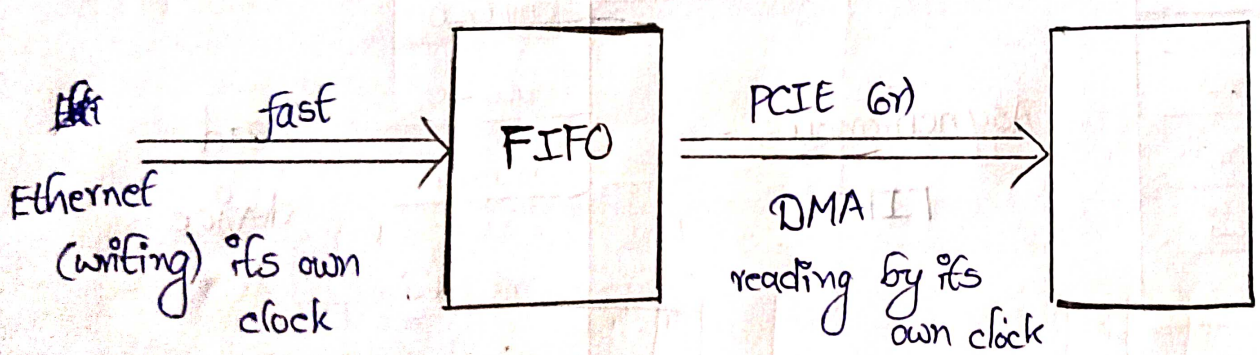
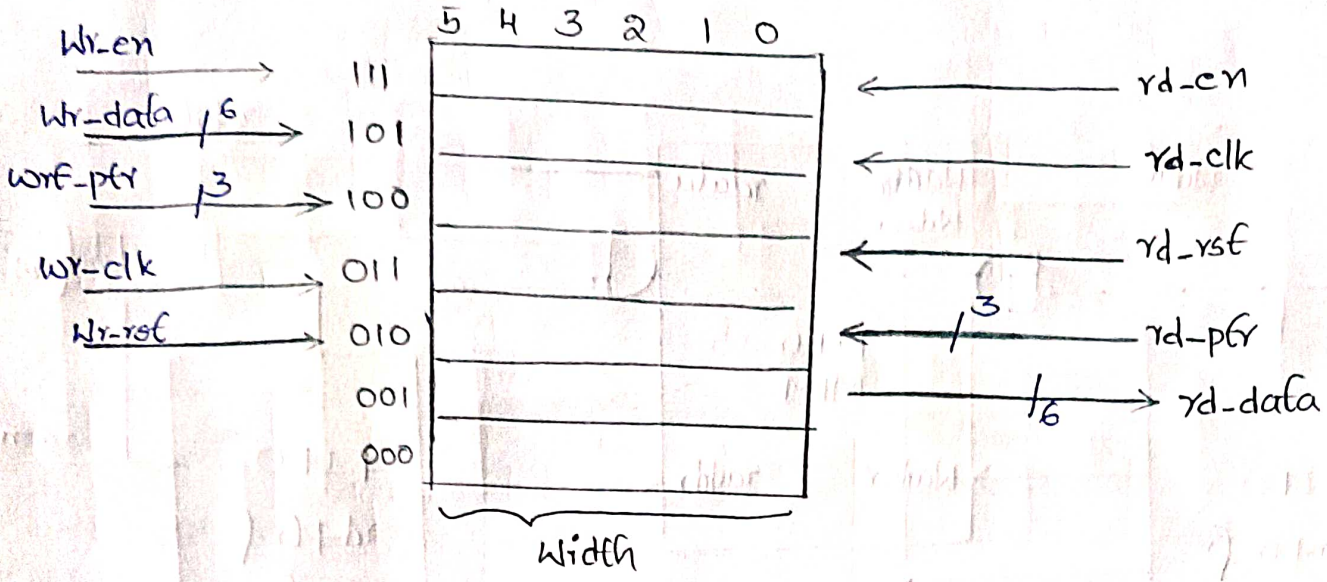
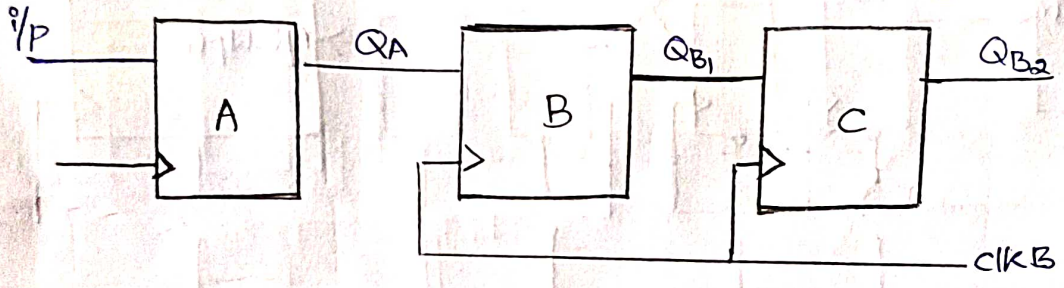
Uses of FIFO:

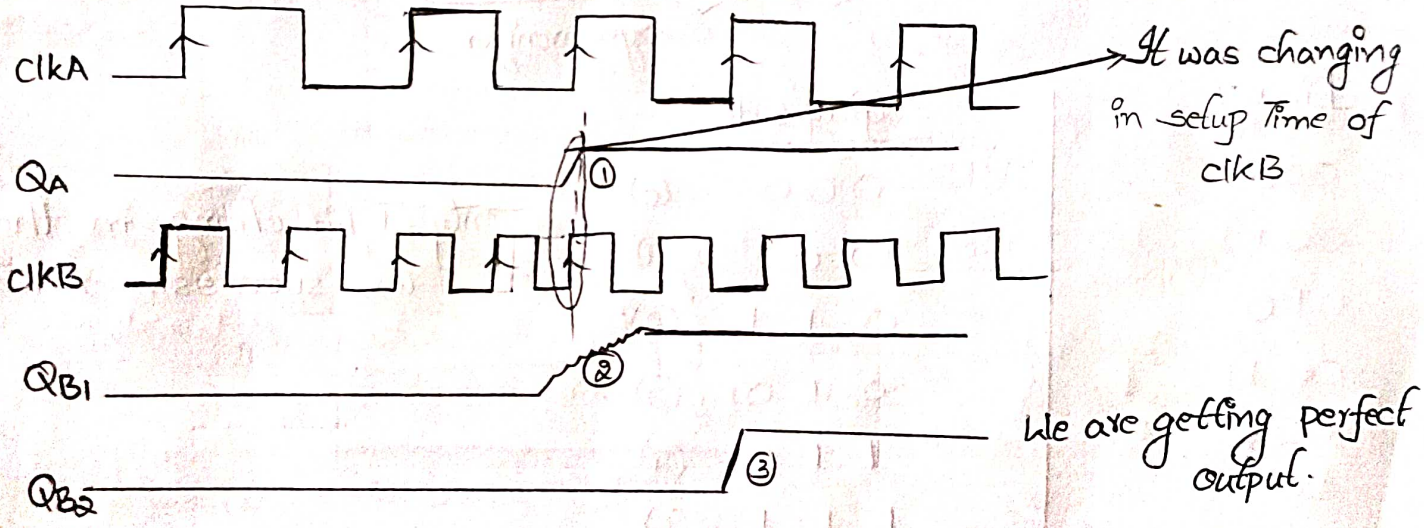
1) As a temporary storage element, where there is a mismatch in data rate coming in & coming out





Synchronizer : It uses to reduce the metastability to a great extent.

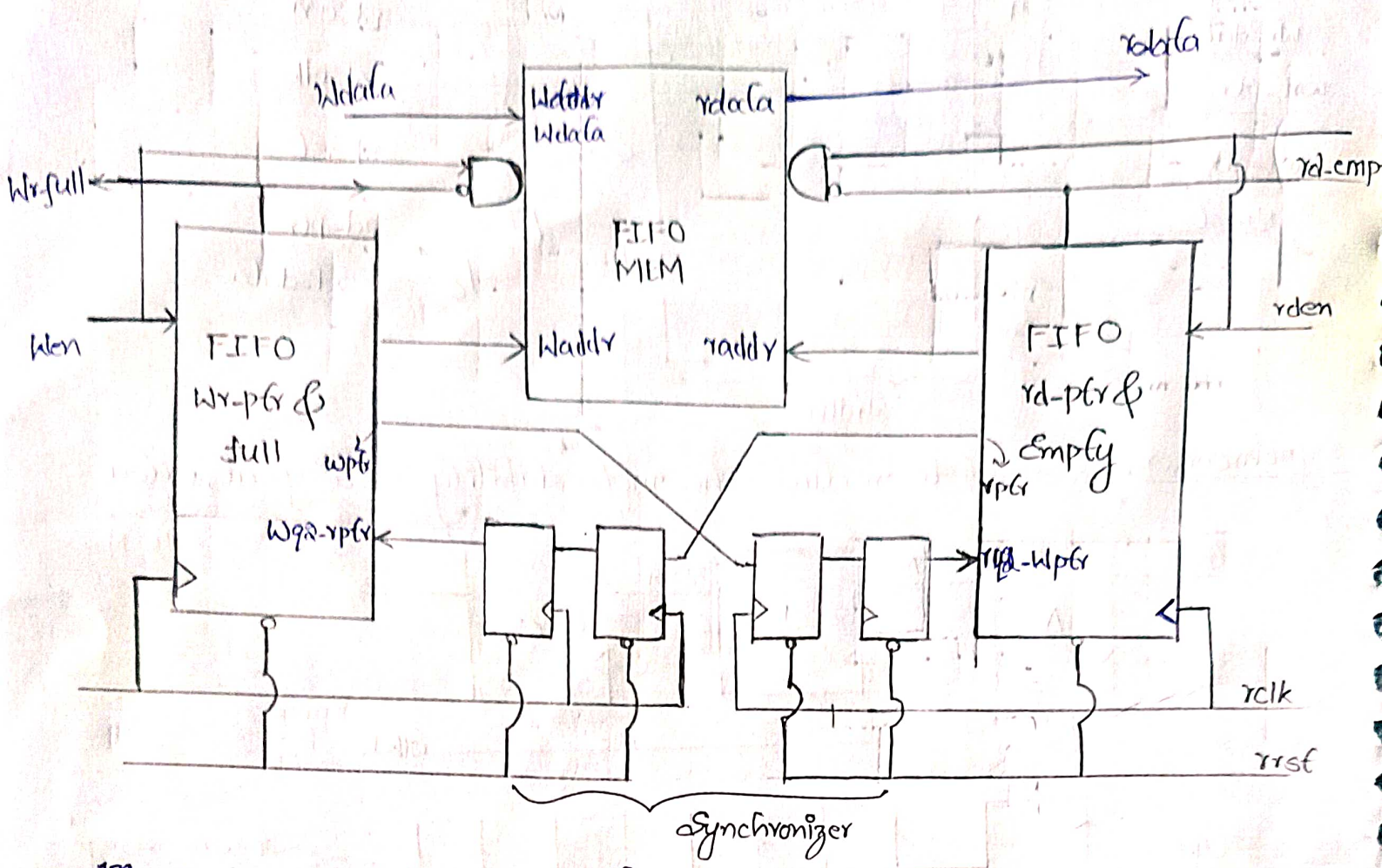




Here the wr_ptr was going to the read module & with help of synchronizer whose clock was ~~wrsp~~ read_clk;

Similarly

rd_ptr was going to the write module & with help of synchronizer whose clock was wrt_clk



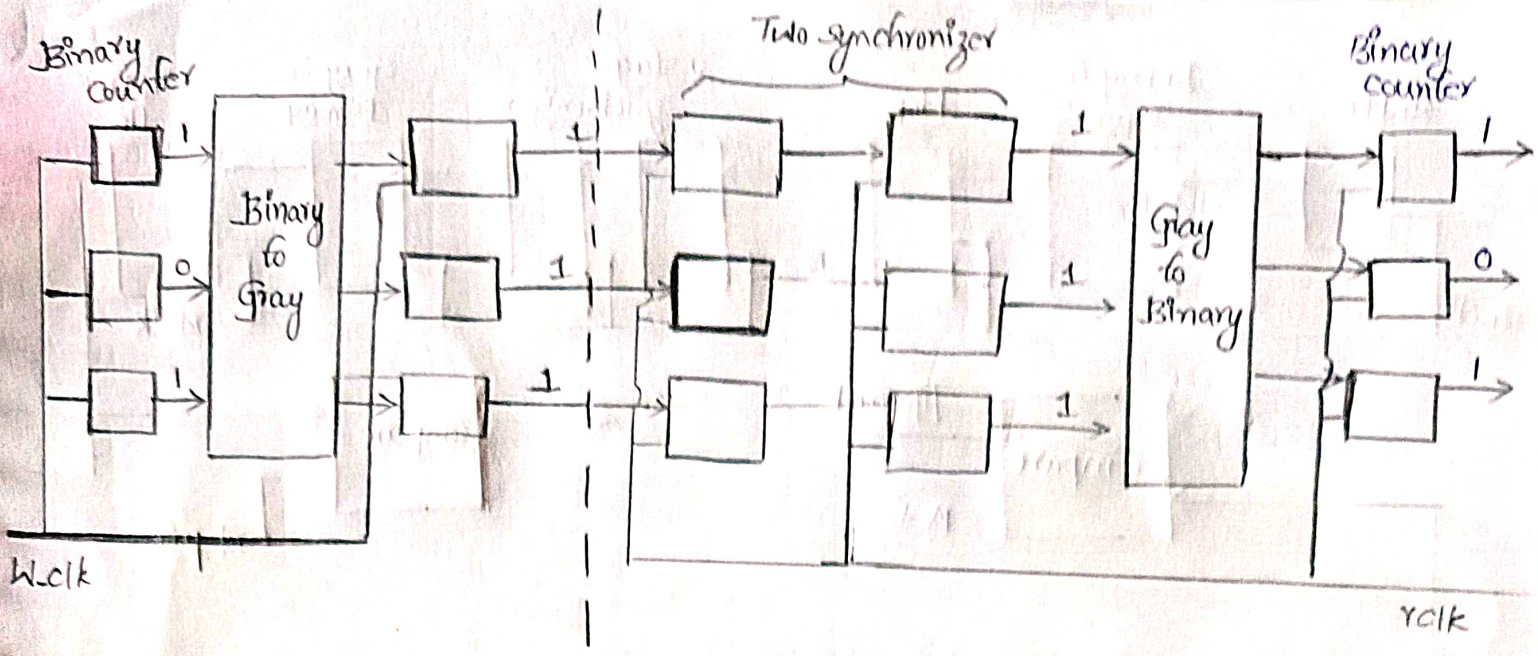
Binary
000
001
010
011
100
101
110
111

Gray Code	
000	(0)
001	(1)
011	(2)
010	(3)
110	(4)
111	(5)
101	(6)
100	(7)

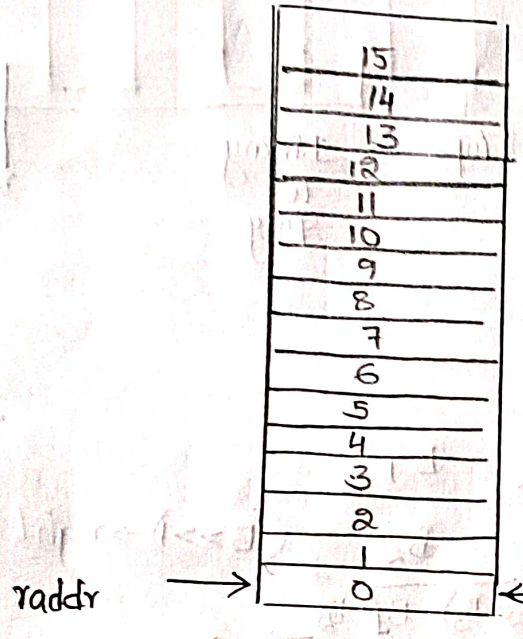
only 1 bit change in the two successive numbers

Decimal	Binary	after synchronization	Binary value
5d	101	101	5d
6d	110	110/111/100/101	5d/6d/4d/7d. false full

Decimal	Binary	Gray	after synchronization	Binary value	
5d	101		~~101~~ 111	111	5d
6d	110		101	111/101	5d/6d



Full & Empty Edition :



Here, rd_addr & wr_addr are pointing to same

ASO, whether our FIFO is full or not We can't say.

→ To avoid this confusion, we will add an extra bit but locations will be same

raddr [3:0] == waddr[3:0]

15	01111	11111
14	01110	11110
13	01101	11101
12	01100	11100
11	01011	11011
10	01010	11010
9	01001	11001
8	01000	11000
7	00111	10111
6	00110	10110
5	00101	10101
4	00100	10100
3	00011	10011
2	00010	10010
1	00001	10001
0	00000	10000

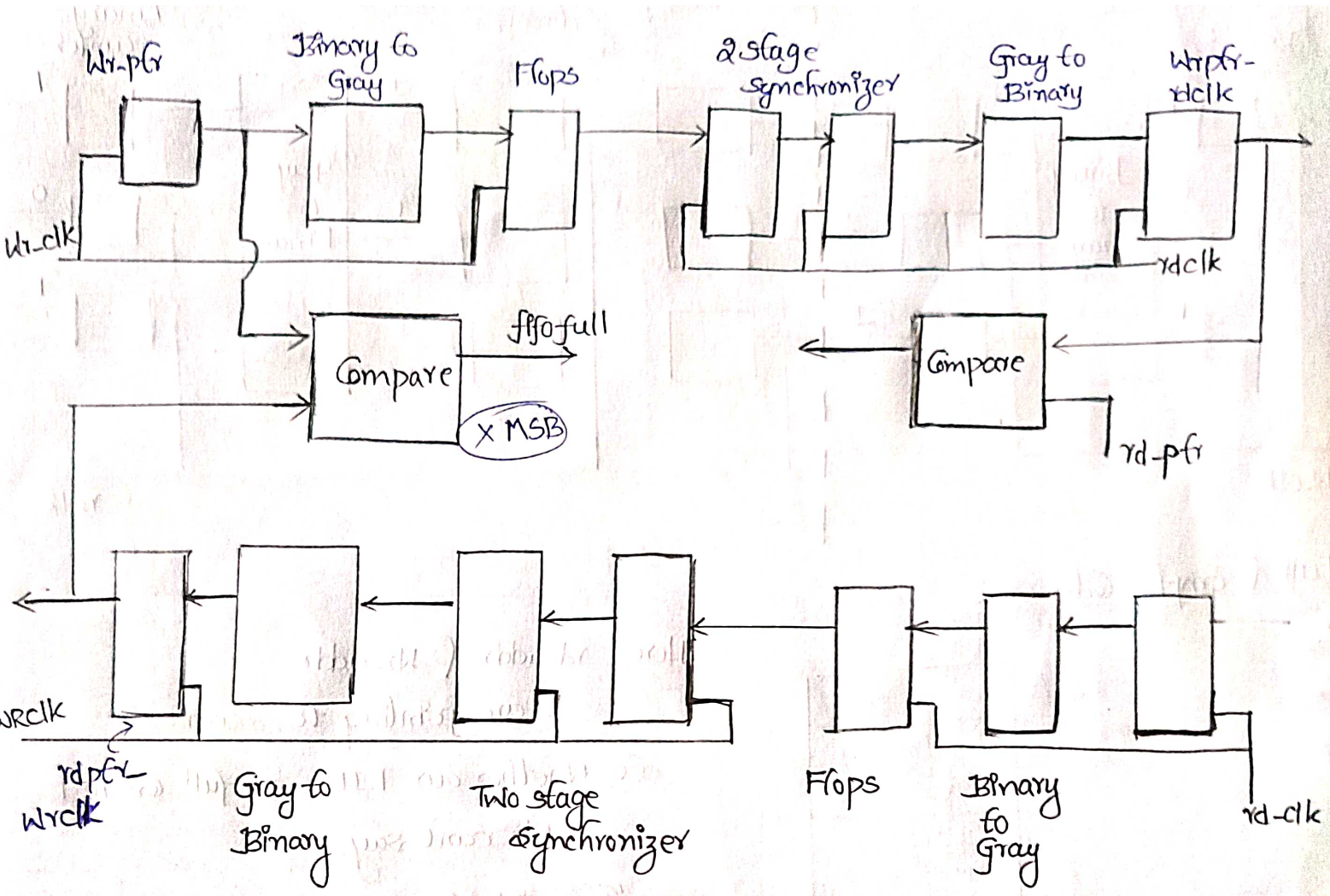
waddr

raddr

Observe the last bit

Full : {~wraddr[4], waddr[3:0]} == raddr

Empty : waddr == raddr



Binary to Gray Code

$B_4\ B_3\ B_2\ B_1$ ⇒ $B_4\ B_3\ B_2\ B_1$

$G_4\ G_3\ G_2\ G_1$ $\quad$ $0\ B_4\ B_3\ B_2$ $\quad$ (B>>1) → right shift

$B_4\ \ B_4\oplus B_3\ \ B_3\oplus B_2\ \ B_2\oplus B_1$ $\quad$ $B_4\ (B_3\oplus B_4)\ (B_2\oplus B_3)\ B_1\oplus B_2$ → Gray code